

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/321353487>

# Power Consumption and Performance Balance (PCPB) scheduling algorithm for computer cluster

Conference Paper · September 2017

DOI: 10.1109/UkrMICo.2017.8095365

CITATIONS

READS

62

5 authors, including:



[Larysa S Globa](#)

National Technical University of Ukraine Kiev Polytechnic Institute

149 PUBLICATIONS 185 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Load management in the heterogeneous data centers of the operator [View project](#)

# *Power Consumption and Performance Balance (PCPB) scheduling algorithm for computer cluster*

Alexander Schill  
Chair for Computer Networks  
Faculty of Computer Science  
Technical University of Dresden  
01062 Dresden, Germany  
[alexander.schill@tu-dresden.de](mailto:alexander.schill@tu-dresden.de)

Larysa Globa, Oleksandr Stepurin,  
Nataliia Gvozdetska, Volodymyr Prokopets  
National Technical University of Ukraine “Igor Sikorsky  
Kyiv Polytechnic Institute”  
Kyiv, Ukraine  
[lgloba@its.kpi.ua](mailto:lgloba@its.kpi.ua), [oleksander.stepurin@gmail.com](mailto:oleksander.stepurin@gmail.com),  
[n.gvozdetska@gmail.com](mailto:n.gvozdetska@gmail.com), [vprokopets95@gmail.com](mailto:vprokopets95@gmail.com)

**Abstract**—The problem of energy consumption is topical in every sphere of human activity nowadays. This problem is especially important for IT (information technologies), where the amount of data that needs to be processed is growing every day. In this paper PCPB (Power Consumption and Performance Balance) algorithm has been proposed. It is an energy-aware scheduling algorithm which is aimed at reducing power consumption of the computer cluster without its performance reducing. The main idea of the proposed approach is to use energy model (as relation between CPU load of the node and its consumed power) for each node in the cluster and find the balance between power consumption and performance while tasks allocating. To compare PCPB with other algorithms mathematical model was developed. The efficiency of the algorithm is proved experimentally: PCPB algorithm allows increasing performance of a cluster and reducing its power consumption at the same time in comparison with widely used Round Robin algorithm.

**Key words**—scheduling algorithm, power consumption, performance, computer cluster.

## I. INTRODUCTION

Data centers are consuming more and more power every year. The reason is the challenge posed by the flood of Web, Social Media, Internet of Things (IoT) and machine-to-machine (M2M). The amount of data that needs to be processed is getting huger and huger.

At the same time servers' sources are used very inefficiently. Statistics got by Google in 2013 show that the servers' load is very unsteady. The average load ranges between 15% and 60% for one server and between 40% and 85% for another [1]. It means that the amount of the consumed power is much bigger than it could be. The more computers are in working state, the more power is being consumed (in idle mode). It is necessary to develop a strategy that will allow processing as many tasks as possible using less servers efficiently.

## II. BACKGROUND

There are different software approaches to energy-aware strategies and algorithms developing. Among them it is possible to distinguish:

- scaling,
- scheduling,
- consolidation approaches [2].

There are also some hardware approaches but they are not being explored in details within this work.

Scaling approach takes into account the load arriving to the computer cluster and adopts the parameters of the processing system in accordance to it. In the work [3] the use of dynamic frequency scaling was investigated and it was achieved 12% reduction in power consumption of the system.

Talking about the energy-aware scheduling algorithms, there are a lot of approaches presented. Their main principle is to allocate each task to the machine that can process it with the least power being consumed but without violating a service-level agreement (SLA) [2].

Consolidation approaches usually mean a runtime migration of virtual machines. Virtual machines migrate in such a way, that some servers in the cluster use as many of their resources as possible and others are switched off.

Among these approaches scheduling was chosen for deeper investigation in this paper, as far as consolidation approach operates with virtual machines environment and scaling approach means direct changing of system's parameters (as processor frequency) in accordance with the load.

Within this research energy-aware scheduling approaches are being analyzed and new energy-aware scheduling algorithm PCPB (Power Consumption & Performance Balance algorithm) has been presented.

In [4] basic algorithms, that are widely used for tasks balancing in server cluster, are being analyzed. The authors propose dividing these algorithms into 3 groups: knowledge-free – there is no information about tasks and sources that are going to be used; energy-aware – taking into account the information about power consumption of each node of cluster; utilization-aware – with information about current CPU utilization of each node. These algorithms are gathered in Table 1.

TABLE 1. BASIC SCHEDULING ALGORITHMS DIVIDED INTO GROUPS

Type	Strategy	Description
Knowledge-free	Rand	Allocates task $j$ to the random node using uniform distribution
	First Available (FIFO)	Allocates task $j$ to the first node that is able to process it
	Round Robin	Allocates tasks $j$ to the nodes one by one
	Least Connections (Min Load)	Allocates task $j$ to the node that has minimal number of connections (is the least loaded)
Energy-aware	Min $t_e$	Allocates task $j$ to the node that has minimal total power consumption at the moment
	Min $e$	Allocates task $j$ to the node that has minimal power consumption at the moment
Utilization-aware	Min $u$	Allocates task $j$ to the node that has minimal level of current load
	Max $u$	Allocates task $j$ to the node that has maximal level of current load
	Min $c$	Allocates task $j$ to the node that processes the least tasks of the same type at the moment

The most widely used scheduling algorithms, that are used nowadays, are Round Robin, Weighted Round Robin, Least Connections.

The simplest and the most common load balancing algorithm within the server cluster is Round Robin [5]. According to this algorithm, all tasks are allocated to all active servers one by one. This algorithm does not require any knowledge about the current state of the servers. But it causes also the main drawback of Round Robin: it is not able to allocate the task to the most appropriate server. Such way of scheduling is simple but not efficient. Round Robin algorithm does not take into account power consumption criteria and possible differences between the nodes of computer cluster.

Another example of widely used scheduling algorithms is Weighted Round Robin. Unlike simple Round Robin algorithm, this modification assigns weighted coefficients of the servers according to their performance. These coefficients are further taken into account while the tasks are being distributed between the processing nodes of the server cluster.

The Least Connections scheduling algorithm directs network connections to the server with the least number of established connections. This is one of the dynamic scheduling algorithms, because it needs to count live connections for each server dynamically. It is best suited for a real server pool where each member node has roughly the same capacity [6].

The algorithms described above do not take into account energy efficiency of computer cluster criteria.

Several energy-aware algorithms are also proposed today. Among them are CTES – Cooperative Two-Tier Energy-Aware Scheduling [7]. Modeling results, that are presented in this work, show that algorithm allows reducing energy consumption of the whole cluster, but the main drawback of the proposed approach is that the authors supposed the model of energy consumption to be linear. Actually such a model shows the real dependency between consumed power and CPU utilization inaccurate.

In our work we propose holding certification of each server in a cluster preliminary to define dependencies  $P =$

$f(\text{CPU})$  to make a decision about every task allocation based on the real data.

Another example of energy-aware algorithm is Min\_c [4]. This strategy takes into account that the tasks that come to the server cluster differ and the resources, required by these tasks, differ too. The main drawback of the proposed approach in paper [4] is that the model of energy consumption is the same for each node. It has nonlinear character, that is closer to reality, but the characteristics of different machines can differ. Therefore, we propose defining  $P = f(\text{CPU})$  dependencies for each machine individually.

To develop any kind of energy-aware scheduling algorithm it is necessary to know the dependencies between consumed power and CPU utilization (model of power consumption).

In paper [8] this model is considered to be linear:

$$P_i = P_{idle} + \sum_r M_i^r \cdot \frac{U_i^r}{C_i^r} \quad (1)$$

where  $P_{idle}$  – idle power consumption of a server,  $M_i^r$  – maximum power consumption values for sources  $r$  matrix,  $U_i^r$  – utility (usage) of sources  $r$ ,  $C_i^r$  – matrix of volumes of sources  $r$  for server.

Linear model is also proposed to be used in work [10]. Such kind of model can be performed as a graph in Figure 1 (a).

In paper [11] the proposed model is nonlinear and shown on the figure 1 (b). It can be described as:

$$P_{idle} + (P_{busy} - P_{idle})(2u_{cpu} - u_{cpu}^r) \quad (2)$$

where  $P_{busy}$  – power consumption of a server with 100% of CPU utilization,  $r$  – adjustment factor that minimizes the mean square error of the model.

In [12] the proposed model is also nonlinear and consists of static  $P_{static}^x(t)$  and dynamic  $P_{dynamic}^x(t)$  power consumption. These characteristics are defined by CPU utilization  $Util_x(t)$ ,

$$COSP = \sum_{x=1}^M \left( \sum_{t=1}^{L_{max}} (P_{static}^x(t) + P_{dynamic}^x(t)) \right) \quad (3)$$

where  $L_{max}$  – maximum length of all tasks schedules, COSP – total sum of power consumptions of all servers in the cluster during all time intervals  $t$ . The model is presented in Figure 1 (c).

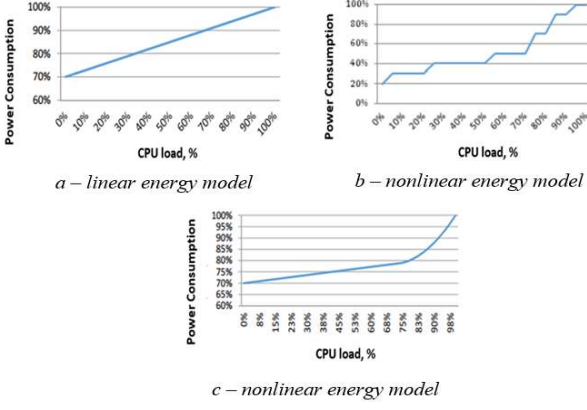


Fig.1.  $P = f(CPU)$  functions of server nodes used for tasks allocation in [10], [11], [12]

In this paper it is proposed to hold certification of each server in a cluster preliminary, store description for each server, evaluate current state of the cluster before the task allocating and allocate task to the server which has the best performance and power efficiency at the same time.

### III. PROBLEM DEFINITION

The main issues that are proposed within this paper are:

- to get individual energy models as  $P = f(CPU)$  functions for each machine;
- to store description about each node of the cluster (volume of RAM, number of processing cores, performance of the node,  $P(CPU)$  function (energy model));
- to schedule tasks taking into account such parameters of each node:
  - availability at the moment;
  - server's performance;
  - server's power consumption.

#### A. Main definitions

Let's consider the task to be a set of operations which need to be processed together in the computer cluster:

$$Task = \{task_i\}$$

Job – is a task with its requirements prepared for processing:

$$Jobs = \{job_i\} \rightarrow \left\{ task_i, \left\{ V_{req_i}, k_{core_{req_i}}, t_{i_{max}} \right\} \right\},$$

where  $V_{req_i}$  - RAM required for the  $task_i$  (maximal value),

$k_{core_{req_i}}$  - number of cores required for the  $task_i$ ,  
 $t_{i_{max}}$  - maximal time for the  $task_i$  to be processed.

Computer cluster – a set of computers which work together and can be viewed as a single system. Node – a unit of the computer cluster, that is performed physically as one computing machine:

$$ComputerCluster = \{N_j\},$$

where  $N_j$  - one of the computer cluster nodes.

FLOPS is a measure of computer performance; a number of operations that can be processed by one processor per time unit [9].

#### B. Formal problem definition

There is a set of the nodes  $\{N_j\}$  in the cluster. Every  $N_j$  node is characterized by:

$V_j$  - volume of RAM available;

$flops_j$  - performance of the  $N_j$  which has  $k_{core_j}$  cores,

where  $k_{core_j}$  - number of the cores on the  $N_j$

$P_j = f_j(CPU_j)$  - power consumption function, defined by  $f_j(CPU_j)$ ,

for  $N_j$  node, experimentally defined;

New task  $task_i$  comes into the system at the moment  $\tau$ .

Each task  $task_i$  requires the set of parameters (predefined):  $\{V_{req}, k_{core_{req}}, t_{max}\}$

$$job_i \rightarrow task_i, \{V_{req_i}, k_{core_{req_i}}, t_{i_{max}}\}$$

It is required to develop a scheduling algorithm so that:

$$P_{\Sigma} \rightarrow \min, t_{task_i} \rightarrow \min$$

where  $P_{\Sigma}$  - the total power consumed by the whole server cluster;

$t_{\Sigma}$  - time required for the set of  $m$  tasks processing.

### IV. ALGORITHM DESCRIPTION

According to the proposed approach each node of the server cluster must be preliminary certified to determine its function  $P_j = f_j(CPU_j)$ . This information should be included into description of each unit with its RAM volume value, number of processing cores and performance (measured in number of floating point operations per time unit (FLOPS)).

The algorithm PCPB, that is proposed in this work, consists of 7 steps. It is described by Figure 2.

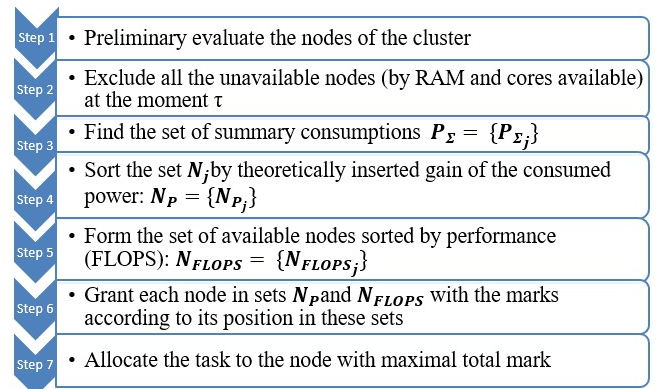


Fig. 2. General description of PCPB algorithm

*Step 1.*

Each node of the cluster should be evaluated preliminary so its description, including  $P(CPU)$  function, should be formed. This procedure should be held once within cluster setup.

New job  $job_i$  comes to the cluster at the moment  $\tau$ .

*Step 2.*

If  $(\Delta V_{j_{avail}} \leq V_{req_i})$  then node  $N_j$  is excluded from the set of available nodes, where  $\Delta V_{j_{avail}} = V_j - V_{j_{used}}$  - volume of RAM available (free) for node  $N_j$ , where  $V_{j_{used}}$  - the RAM used at the moment  $\tau_{k-1}$ ;

If  $(\Delta k_{core_j_{avail}} \leq k_{core_{req_i}})$  then node  $N_j$  is excluded from the set of available nodes, where  $\Delta k_{core_j_{avail}} = k_{core_j} - k_{core_j_{used}}$  - number of cores, available on each node  $N_j$ ,

where  $k_{core_j_{used}}$  - the number of used cores on the node  $N_j$  at the moment  $\tau_{k-1}$ ;

Thus:  $\{N_{avail_j}\} \in \{N_j\}$ ,

*Step 3.*

For each node  $N_j$  the relations between its load (CPU) and consumed power are measured. An example of  $P_j = f_j(CPU_j)$  function is presented in Figure 3.

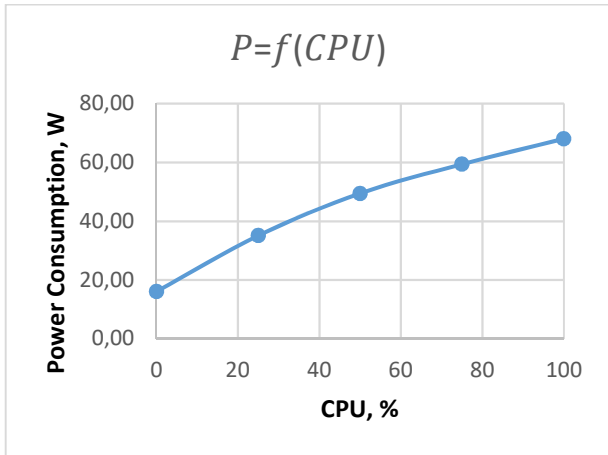


Fig. 3. One of the functions  $P_j = f_j(CPU_j)$  gotten experimentally

$\Delta P_{j_i}|\tau_k = f_j(CPU_{j_k})$  - is a gain of the power consumption produced by the  $job_i$  processed by node  $N_j$

Supposing that the  $job_i$  is given to the processing to the node  $N_j$ , the power, theoretically consumed by the whole cluster, can be defined as:

$$P_{\Sigma_i}|\tau_k \& N_j = P_{\Sigma}|\tau_{k-1} + \Delta P_{j_i}|\tau_k$$

Thus, the set  $P_{\Sigma} = \{P_{\Sigma_i}\}$  is formed by theoretical placing the  $job_i$  to each of the nodes  $N_j$ .

*Step 4.*

Form the set of nodes, sorted by performance from minimum to maximum, from the set of available nodes:

$$N_{flops_{avail}} = \{N_{flops_{j_{max}}}, \dots, N_{flops_{j_{min}}}\}$$

*Step 5.*

Thus, the set of available nodes  $N_{avail}$  is characterized by 2 parameters - flops and power consumption in case of  $job_i$  placement on this node  $\Rightarrow N_{avail} = \{N_{avail_s}(flop_s, P_{\Sigma}|\tau_k \& N_s)\}$

The set  $N_{avail}$  is sorted from min to max by criteria power consumption. Thus,  $N_{avail_p}$  is got.

The set  $N_{avail}$  is sorted from max to min by performance criteria (there is no need to re-sort this set after each task coming). Thus,  $N_{avail_{flops}}$  is got.

*Step 6.*

It is necessary to find an intersection between these sets using special strategy based on the worth variants excluding and intersection searching from the subsets got

Two sorted sets  $N_{avail_{flops}}$  and  $N_{avail_p}$ , containing several common elements (nodes), form the set of suitable decisions  $Dec$

*Step 7.*

The task is allocated to machine that has the best characteristics of performance and energy efficiency in such way:

Each machine from the sorted set  $N_{avail_{flops}}$  and  $N_{avail_p}$  gets its mark in range from 1 to n;

The mark given for performance is summed with the mark given for energy efficiency;

The task is allocated to the machine that has the highest mark.

The principle of marks giving is shown in Figure 4 (an example for 5 machines).

Sorted set by performance		Sorted set by energy efficiency	
Node	Mark	Node	Mark
Node #3	5	Node #1	5
Node #1	4	Node #5	4
Node #5	3	Node #2	3
Node #2	2	Node #3	2
Node #4	1	Node #4	1

Node	Total mark
Node #1	9
Node #2	5
Node #3	7
Node #4	2
Node #5	7

**=> The task is allocated to the node #1 as it has the best mark 9**

Fig. 4. An example of giving marks and tasks allocating for 5 nodes in cluster

The algorithm was implemented by the code of shedular written in C programming language.

## V. EXPERIMENT

To evaluate PCPB's performance in real environment an experiment was carried out. For comparison the Round Robin algorithm was chosen as a simple and widely used algorithm.

For the experiment five nodes were used. Among them, three belonged to the same type (had similar characteristics) and two belonged to the second type. Characteristics of each type of nodes are shown in Table 2.

TABLE 2. CHARACTERISTICS OF CLUSTER NODES USED IN EXPERIMENT

	Nodes of type 1 (x3)	Nodes of type 2 (x2)
RAM volume	4 Gb	8 Gb
Processor type and frequency	CPU Intel Core 2 Quad Q9400 2.66GHz	CPU Intel Core i5-43 3GHz
Performance	35.7GFLOPS	40-50GFLOPS

These 5 computers were joint in a server cluster. One computer was selected as the main node, it operated as a scheduler. Other subordinate nodes were equal to each other and had a direct connection with the main node (star topology).

To evaluate Round Robin and PCPB algorithms the same packages of tasks were used. The tasks within the package were different. As for task,  $\pi$  number calculation up to 1000, 5000, 10000, 25000, 50000, 75000, 100000 decimal places was chosen. Number of decimal places was normally distributed random variable.

According to the proposed approach, servers should be certified before calculations ( $P_j = f_j(CP_j)$  functions should be defined). To define these functions, each server was in serial loaded on 25, 50, 75, 100% (with the help of stress test). The values were approximated by the polynomials of the 4th degree. Derived functions can be seen in Figure 3, where the first number of node is its type (of two types, described above), the second is the number of node.

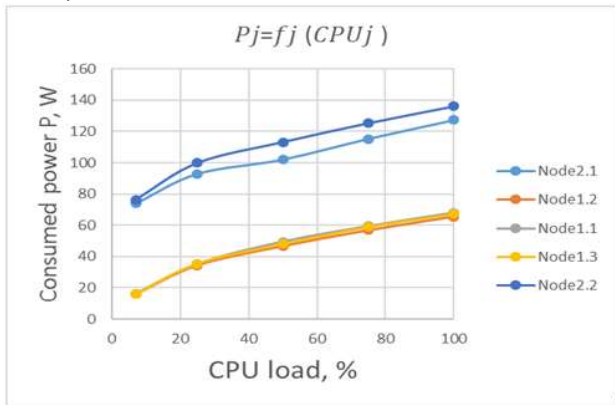


Fig.5.  $P_j = f_j(CPU_j)$  functions gotten experimentally

After getting analytical functions for each node, evaluation of Round Robin scheduling algorithm was carried out. The results of experiment are shown in Table 3 (the values in the table are arithmetic average of 20 measurements. Consumed power is the total power that was consumed by the whole cluster while the given package of tasks was processing).

After evaluating Round Robin algorithm, the code of scheduler was changed. All conditions including tasks packages remained the same. The experiment for PCPB algorithm was carried out. The results of this evaluation are also presented in Table 3.

TABLE 3. EXPERIMENTAL PCPB & ROUND ROBIN COMPARISON FOR 5 NODES IN CLUSTER

Algorithm	Average time of 1 task processing (s)	Power consumed (W)	Gain (in comparison with Round Robin) (%)	
			in time	in power consumed
Round Robin	14.835	376.88	0%	0%
PCPB	13.324	365.8	10,2 %	3%

From the tables below it is possible to see that the proposed algorithm PCPB is a bit more efficient than Round Robin according to both criteria. The gain in performance reaches 3% and the gain in energy efficiency – 10,2% in comparison with Round Robin algorithm..

Not so big efficiency is explained for such a reason that only 5 nodes were used in experiment. To test PCPB's efficiency for more nodes in cluster mathematical modeling was used.

## VI. MATHEMATICAL MODELING

Before carrying out the modeling a real computing cluster, containing 5 nodes, was certified (functions  $P_j = f_j(CP_j)$  were defined).

To evaluate an efficiency of the proposed approach mathematical modeling was held. To carry out the modeling MATLAB tools were used. To compare power consumption and speed of the proposed algorithm with others, such algorithms were chosen as:

- 1) First available (FIFO) (as the simplest one);
- 2) Round Robin (RR) (as simple and widely used algorithm);
- 3) Round Robin with weighted coefficients (as improved version of RR);
- 4) Least Connections (also widely used in real environment).

To make sure that MATLAB modeling gives statistically equivalent to the real experiment results, modeling for the same (as for real experiment) conditions (5 nodes with the same parameters) was carried out.

Taking into account this issue, the modeling was carried out according to the following plan:

- 5) Modeling for 5 machines (according to the real experiment, described in paragraph 5);
- 6) Modeling for 20 machines;
- 7) Modeling for 20 machines with typical datacenter working day imitation.

### A. Input parameters defining

To evaluate the algorithms processing the same for each algorithm package of tasks was used to be uploaded to the modeled servers.

Parameters of the tasks:

- Number of input tasks: 1750
- Time moments of the tasks' appearing: 0...3600 sec
- Memory amount required by each task: 100 Mb...1 Gb
- Number of cores required by each task: 1...4
- The volume of each task, as a number of floating-point operations to be carried out: 10...500 GFLOPS

The tasks appear in the system randomly (without any predefined sequence).

### B. Modeling in case of 5 nodes in the computer cluster

Parameters of the computer cluster nodes (according to the real hardware used for experiment (paragraph 5)):

- Number of nodes in cluster: 5

	Nodes of type 1 (x3)	Nodes of type 2 (x2)
RAM volume	4096 Mb	8192 Mb
Number of cores	4	4
Performance	35.7GFLOPS	40 GFLOPS

- Function of power consumption P(CPU) in the form of polynomial as:  
 $a * x^4 + b * x^3 + c * x^2 + d * x^1 + e * x^0$   
for each node (defined preliminary).

The results for cluster consisting of 5 nodes are presented in Table 4.

TABLE 4. MODELING FOR 5 NODES IN CLUSTER RESULTS

Algorithm	Average time of 1 task processing (s)	Power consumed (W)	Gain (in comparison with Round Robin) (%)	
			in time	in power consumed
First available (FIFO)	13.146	371.61	12,8%	1,4%
Round Robin	14.835	376.88	0%	0%
Least Connections	13.483	375.64	9,1%	0,3%
PCPB	12.824	364.6	13,6%	3,3%

These results allow us to suppose, that it is possible to trust MATLAB modeling in this case. The gain

received in real experiment and within the modeling is statistically equivalent.

In this case we see that the gain in performance for PCPB reaches 13,6% and the gain in energy efficiency – 3,3% in comparison with Round Robin algorithm. In comparison with First Available algorithm that also shows good results by both criteria PCPB has the gain in performance 2,4% and in energy efficiency – 1,5%.

### C. Modeling in case of 20 nodes in computer cluster

Parameters of the computer cluster nodes:

- Number of nodes in cluster: 20
- Volume of the RAM for each node<sup>1</sup>: 4096, 16384 or 8192
- Number of cores for each node: all 20 nodes have 4 cores
- Performance of each node<sup>1</sup>: 35.7, 45.7, 75, 25.7, 64.7 or 30.7 GFLOPS
- Function of power consumption P(CPU) in the form of polynomial as:

$$a * x^4 + b * x^3 + c * x^2 + d * x^1 + e * x^0$$

for each node (different coefficients of polynomial were used).

The results for 20 nodes are presented in Table 5.

TABLE 5. MODELING FOR 20 NODES IN CLUSTER RESULTS

Algorithm	Average time of 1 task processing (s)	Power consumed (W)	Gain (in comparison with Round Robin) (%)	
			in time	in power consumed
First available (FIFO)	11.636	1436.1	25.2%	4.5%
Round Robin	15.55	1500.8	0%	0%
Least Connections	13.01	1455.9	16.38%	3%
PCPB (Proposed)	10.89	1376.3	<b>30%</b>	<b>8.3%</b>

It is possible to see that in case of 20 nodes in cluster the gain in performance for PCPB is much bigger and reaches 30% and the gain in energy efficiency is also bigger and reaches 8,3% in comparison with Round Robin algorithm. In comparison with First Available algorithm that also shows good results according to both criteria PCPB has the gain in performance 6,4% and in energy efficiency – 4,2%.

<sup>1</sup> combining RAM volume values with different possible performance values (from the given sets) 10 different types of machines are got

D. Real working day (24 hours) of computer cluster containing 20 nodes modeling

To model the real working day of the computing cluster containing 20 nodes load timing from [13] was used. It is presented in Figure 6.

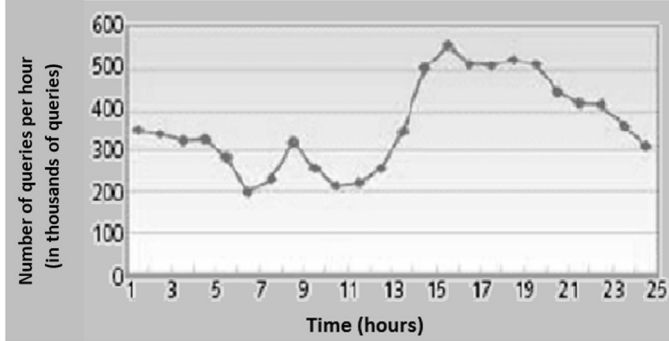


Figure 6. Load timing in datacenter during the working day

Parameters of computers and tasks were the same. The results of one working day of server cluster, containing 20 nodes, are presented in Table 6.

Table 6. Modeling of 20 nodes in cluster working the whole day results

Algorithm	Average time of 1 task processing (same tasks) (s)	Power consumed (W)	Gain (in comparison with Round Robin) (%)	
			in time	in power consumed
First available (FIFO)	11.71	29725	34,58%	2,86%
Round Robin	15.76	30578	0%	0%
Least Connections	12.91	29739	9,1%	2,74%
PCPB (Proposed)	10.07	28568	<b>36,1%</b>	<b>6,57%</b>

In case of 20 nodes in cluster working during the whole day the gain in performance for PCPB reaches 36,1% and the gain in energy efficiency reaches 6,57% in comparison with Round Robin algorithm. In comparison with First Available algorithm that also shows good results according to both criteria PCPB has the gain in performance 14% and in energy efficiency - 3,9%.

In Figure 7 power consumption graph of the whole cluster during the workday is shown (PCPB is named as proposed).

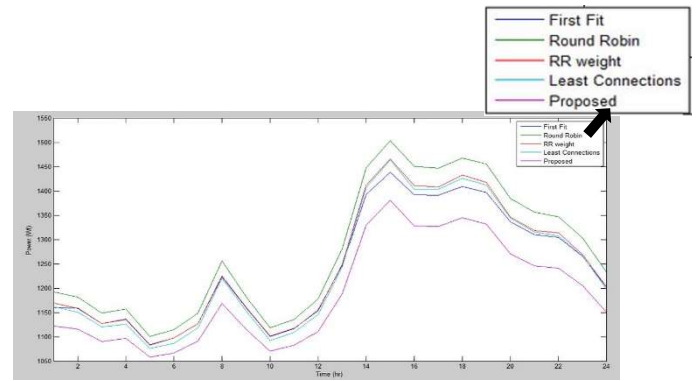


Figure 7. Power consumption of cluster during the workday graph

As it is possible to see on the graph, the best results were presented by PSPB algorithm.

The average time of task processing is the second evaluated parameter. The results of time efficiency are shown in Figure 8.

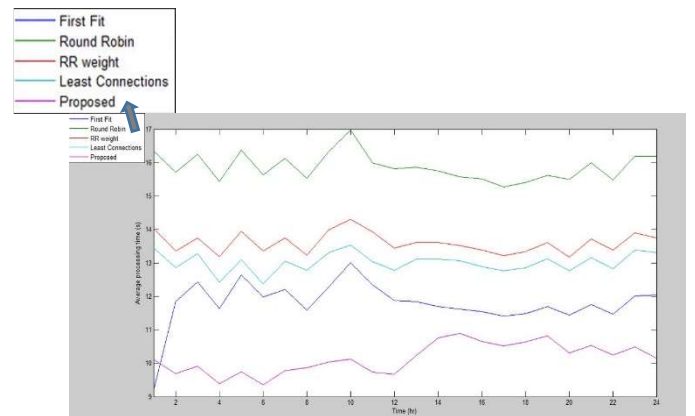


Figure 8. Average time of 1 task processing by cluster during the workday graph

As it is seen from the graphs, PCPB shows the best results in all respects - energy and power efficiency.

VII. FUTURE WORK

As for future work it is planned to develop modifications to the proposed algorithm. The main ideas of modifications are the following:

1. to divide tasks and nodes into groups ("hard" tasks should be allocated only to those nodes that have better performance);
2. to drive some nodes with worse parameters into sleep mode if the load of a cluster is not high.

It is necessary to test PCPB algorithm and its modifications in real environment, so, it is planned to carry out an experiment in a real datacenter.



## VIII. CONCLUSIONS

In this paper PCPB algorithm was proposed. It is an energy-aware scheduling algorithm which is aimed at reducing power consumption of the server cluster without its performance reducing.

The proposed algorithm was compared with existing approaches using mathematical modeling. To prove the efficiency of PCPB an experiment was carried out. The results of the experiment have shown that the proposed algorithm gives benefits in power consumption and in performance as well. The gain of PCPB reached 30% in average processing time and 8,3% in power consumption (in case of algorithm performance in cluster consisting of 20 nodes evaluation).

## REFERENCES

- [1] L. A. Barroso, J. Clidaras, and U. Hoesle "The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines", 2nd ed. San Rafael, CA, USA: Morgan & Claypool, 2013.
- [2] Christoph Möbius, Walteneus Dargie, Senior Member, IEEE, and Alexander Schill "Power Consumption Estimation Models for Processors, Virtual Machines, and Servers", IEEE Transactions on Parallel & Distributed Systems, 2014
- [3] W. Dargie and A. Schill "Analysis of the Power and Hardware Resource Consumption of Servers Under Different Load Balancing Policies," in Proc. IEEE 5th Int'l Conf. CLOUD, 2012, pp. 772-778. [Online]. Available: <http://dx.doi.org/10.1109/CLOUD.2012.30>
- [4] Fermin Armenta-Cano, Andrei Tchernykh, Jorge M. Cortés-Mendoza, Ramin Yahyapour, Alexander Yu. Drozdov, Pascal Bouvry, Dmitry Kliazovich, Arutyun Avetisyan "Heterogeneous Job Consolidation for Power Aware Scheduling with Quality of Service", Russian Supercomputing Days, Moscow, 2015
- [5] [https://en.wikipedia.org/wiki/Round-robin\\_scheduling](https://en.wikipedia.org/wiki/Round-robin_scheduling)
- [6] <http://www.ijcit.com/archives/volume4/issue2/Paper040211.pdf>
- [7] S. Hosseinimotlagh, F. Khunjush, and S. Hosseinimotlagh, "A Cooperative Two-Tier EnergyAware Scheduling for Real-Time Tasks in Computing Clouds," in Proceedings of the 2014 22Nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, Washington, DC, USA, 2014, pp. 178-182.
- [8] A Transient Model for Data Center Thermal Prediction  
Michael Jonas, Rose Robin Gilbert, Joshua Ferguson, Georgios Varsamopoulos, Sandeep K. S. Gupta Impact Lab, <http://impact.asu.edu/> School of Computing, Informatics and Decision Systems Engineering, ASU, Tempe, Arizona
- [9] <https://ru.wikipedia.org/wiki/FLOPS>
- [10]. A. Beloglazov, J. Abawajy, and R. Buyya, Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing, Future Gener. Comput. Syst., vol. 28, no. 5, pp. 755-768, May 2012.
- [11] Power Provisioning for a Warehouse-sized Computer Xiaobo Fan Wolf-Dietrich Weber Luiz André Barroso Google Inc. 1600 Amphitheatre Pkwy Mountain View, CA 94043 {xiaobo,wolf,lui}@google.com
- [12]. Y. Gao, Y. Wang, S. K. Gupta, and M. Pedram, An Energy and Deadline Aware Resource Provisioning, Scheduling and Optimization Framework for Cloud Systems," in Proceedings of the Ninth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, Piscataway, NJ, USA, 2013, pp. 31:1-31:10.
- [13] <http://www.osp.ru/os/2004/02/183912/>